

ACE **ATARI**
COMPUTER
ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

JUNE, 1983

Mike Dunn & Jim Bumpas, Editors

SPECIAL TUESDAY MEETING
ATARI CAMERA CREW HERE!

BENIOFF IS BACK!

Benioff At Large

The Return Of The Jenioff: "Look, Marc, take my word for it: You are not going to have time to write articles while you are going to USC.", chants the intelligent editor to the bewildered writer. Little did I know, how right Dr. Dunn was. Over the past 8 months, I have written a little here, and a little there — but nothing like the regular schedule I was keeping before. This article marks my two year anniversary with ACE. So, now I am out of school for the summer, I decided to sit down and write the article waiting to be written.

Say Good Bye! No, you are not going to say good bye to me, but to the following ATARI products. Say good bye to the: 400, 410, 800, 810, 822, 825, and the 850. According to high sources these products will be discontinued at the JUNE CES. But, there will be some products to replace them. The following products will be added at the CES: 600XL, 800XL, 1201XL, 1022, and 1027 Letter Quality Printer. So, what are all these things?

The New Products: The 600XL is a 16k full stroke 400. It is expandable to 64K through a new board. It will have many of the same features as the 1200XL. The 600XL will retail for about \$149.00. The 800XL is 64K 800 in a black and silver C64-type case. This new machine will retail for about \$299.00. The 1201XL is supposedly a 1200 with a built in modem and a speech synthesiser. It also should be expandable to carry a Sony 3.5" drive. The 1022 is a dot matrix printer, and the 1027 is letter quality. Atari is also planning to replace the 810 with a double/single density Tandon drive. All of these new hardware products sound very interesting.

New Software From ATARI: In conjunction with the 5200 system, ATARI will be incorporating the following software to the PCS line: Pole Position, Joust, Tempest, Battlezone, Pengo, Moon Patrol, Kangaroo, Robotron, Jungle Hunt, and Vanguard. Pole Position and Vanguard have been converted for the PCS and should be shown at the show. Without adaptations for the 5200, these new software products will be released: Microsoft Basic 2, DOS 3, Dig Dug, Assembler Editor 2, New Basic, Eastern Front Rom, Donkey Kong, Tennis, Logo, MS. PacMan, Superman 3, Music System 1 + 2, and a new game called Liberator. All of these new software products will be welcome additions to anyone's ATARI library.

A Coleco VCS? Evidently, ATARI has sold a license to Coleco to produce a VCS clone. The new machine called Gemini, will retail for about \$75.00. But, look out, because ATARI is going to turn around and announce a new VCS for \$50.00. That will not make Coleco too happy.

New Third Party Software: The following products have recently been released for the ATARI PCS. I am unable to review all of these products, but have rated them with stars. (5-Great...1-Not So Good)

Jawbreaker II, ****
Wall War, ****, Sierra
AE, ****, Sierra
Sky Blazer, ***, Broderbund
Shamus: Case II, *****, Synapse
Poker Sam, ***, Don't Ask
Spy's Demise, ****, Penguin
Pharaoh's Curse, *****, Synapse
Zaxxon, ****, Datasoft
Matchboxes, ****, Broderbund
Ultima II, ****, Sierra
Suspended, *****, Infocom

Gwendolyn by Marc R. Benioff, the stars would run off the page, Artworx

Game Hints: Have you ever tried: 1)type the phone number of Big Five into the game during game play. Then press SHIFT and the level you want to go to. 2)Press CTRL G in Chicken. 3)Put the joysticks in port 3 + 4 in Sea Dragon and select your own level. 4)Press OPTION after being killed in any of the new EPYX games.

A New ATARI? ATARI does not sit around and do nothing all day, they develop new things for future years. One of the newest things is a new computer called a mountain range and also a popular software house (Can You Guess It's Name?). This computer will have 5 special processors on board and a megabyte of on board memory. It is the size of a book, and the screen lifts up, in a similar way to a book cover. It is a 4 color liquid crystal display. The music system (called by a popular composer's name, Ama-who?), is superb. It is planned for release sometime in 1984. I can hardly wait for this one.

Me And My 1200. Recently, I bought a 1200, and I don't regret it for a second. I think some of you have also bought some, and you may be interested in little tricks. Here is one: Poke 622,255. OP in memory. The listing FINE SCROLLS upward. This is a nice feature of the 1200.

The end. And so I come To the end of another Benioff At Large. And, I didn't use awesome, once! That was beyond Awesome!

—Marc

News and Reviews

by Mike Dunn, Co-Editor

First, the rumors about the new line-up of Atari computers, according to various other newsletters, club presidents, and others who make guesses — let's see how much comes to pass. First the low end will be the Atari 400, which will sell for \$99 to compete with the "toy" computers. Next will be the Atari 600, which looks like the 1200 but has 16K. It is being advertised by one mail order outfit for \$199. It will have an optional outboard interface allowing expansion of memory to 64K plus serial and parallel ports. Next will be a stripped down Atari 800 with 48K but no right slot or extra slots for new boards, and only two joystick ports. The next upgrade will be a redesigned Atari 1200 allowing all software to work with it — new ROMs, or perhaps just the 800 ROMs. Next in line will be the present Atari 800 — by the way, you can now get a \$100 rebate from Atari if you buy one — a truly great buy. The top of the line is a new computer, name unknown "that will make all the user club members who were disappointed with the 1200 leap with joy". To be announced in June at the CES, it will feature a 16-bit processor, or possibly the 8-bit version of the 68000, and have a 6502 co-processor so present Atari software can be used. It will have expansion slots, and all the goodies you might want. We will see if any of the above happens!

In the March issue of Oregon Business, there is a wonderful article of a 72 year old inventor who has lived most of his life in a small coastal Oregon town. Ted Smith developed an ink for printer ribbons after he decided it was too expensive to buy ribbons for his TRS-80 computer. Starting a company in his home, using handicapped persons when possible, he now markets his MacInker[re-inkers nationally. Since I use a new ribbon for each issue to get the sharpest reproduction, I gave Mr. Smith a call. On the other end of the line was a delightful, enthusiastic gentleman who explained he had a model for over 100 printers now, at list prices of \$52 to \$64. I ordered one, and he said to use it, and if I liked it, send me a check! After explaining about our club, and that our membership was world-wide, etc., he said he would give a 25% discount to ACE members if they just tell him!! The next day, I received my new MacInker for the IDS MicroPrism, a printer which is difficult to find ribbons for anywhere except the company. The device is a large, oval table made of cast aluminum, with holes drilled into it to hold your ribbon cartridge. It hooks up to a slow speed motor, and winds between two felt reservoirs filled with the provided ink. You then let it run for 1/2 to 1 hours, and allow the ribbon to dry overnight. Your "new" ribbon is really "better than new" as the advertisement says, as it has not been able to dry out in the package. The unit is not very pretty — it looks like it was designed by a machinist for function and durability (Mr. Smith designed and manufactured reloading equipment for target rifleman before he retired), but it works great! It is a little messy, so wear rubber gloves or have "Goop" available to clean up with. Contact Mr. Smith at S.A.S. Industries, 3091 N. Bay Dr., North Bend, Oregon 97459 (503) 756-7155 — be sure to mention ACE for your discount.

Just at press time, I received several new things. First is a new book from Creative Computing, called **The Creative Atari**. Edited by David and Sandy Small and George Blank, it combines much of what has been in the Creative Computing issues, as well as much new and expanded articles. It even has an article by me which I submitted to the magazine about 2 years ago, and was told it would be in their new book, "due out in a couple of months". Many of the new articles are by editors and presidents of many of the user groups around the country — a very nice book, looks like one for the "Must" list. Broderbund sent a wonderful family game called **Matchboxes**. It really looks great — based on the game of the same name, it has spectacular graphics and various word game choices. We'll find some young person to review it. I have also bought the OS board with David Young's Omnimon! monitor, and the MPC EPROM burner. On order still is the new Austin Franklin 80 — column board, due to be shipped "any day now" for the last month. Hope to get it to work with my ATR 8000. Speaking of the ATR 8000, local club members are buying them like crazy, and the program allowing you to read many disk formats is about ready to be shipped.

Our new disks, the "Best of 1983 #1" which includes the GTIA Sketchpad and some pictures, several Stan Ockers games, games by Sydney Brown, etc., and "Source Codes", a collection of Assembly language routines are now available for \$15 each or \$20 for both. There are also 3 new education disks, including some modified old programs and many new ones for \$10 each or \$15 for 2. These are also all available on cassette for the same single price.

Chuck and Jody Ross, our librarians, are completely revising and updating our **Library List**. It will be better organized, better identified, and easier to understand. Send a SASE to them at 2222 Ironwood, Eugene, Or 97401 for the new list, and they will send it when completed.

BUMPAS REVIEWS

The Creative Atari, ed. by David Small, Sandy Small and George Blank (Creative Computing Press, 1983) \$15.00 (?).

The preface of this book tells you you have missed out on some of the best material ever published on the Atari 400/800/1200 computers if you are not already familiar with **Creative Computing** magazine. Of course, they are not going to tell you that without adding the reassurance this book will give you. Many articles from CC are reprinted here, along with new material. The preface states the philosophy: "learning by doing is the best way to learn."

The book describes itself as an "Atari Almanac". I agree. It is an almanac. Most of the articles and sections in the book are designed to help the reader get the most out of the Atari. There are tutorials, reviews, hardware projects, and many programs. The program listings are mostly demos, but there are some user programs, too.

A couple of items are a bit dated — especially where the writer speculates upon new Atari computers such as the "1000", or upon the prospected need for a do-it-yourself light pen project since \$100 is too much to pay. But I have no complaints.

The book is divided into 5 Parts and an Appendix, totalling nearly 250 8.5 x 11" pages. Parts I and II concern themselves with Graphics (a tutorial and tips). Here you will find several demo programs to show off some of the better characteristics of Atari graphics.

Part III has over 100 pages and contains mostly hardware and software reviews. But here you will also find a half a dozen hardware projects. This section also contains several more demonstration programs on sound capabilities, discussions for programmers on string handling and other unique features of Atari BASIC, questions and answers, and a description about how to run a serial printer with the Atari. The review of the Mosaic 64k RAM card suggests possibilities for its use which almost impell me to go out and buy one now.

Part III also includes the review by Mike Dunn on BASIC A+.

Part IV is a disk drive tutorial which will be of interest to most users who aren't sure just how the gadget works when it talks to our Atari. The last section contains 5 user program listings in its 30 pages. They have been chosen to demonstrate handling of various programming techniques. So these listings can be used as a learning tool in themselves. I can't comment on their quality as games yet, but there are at least two of them I plan to type in.

I've already typed in the entire "Sound Library" from Part III. They are very good. These are little 1 or 2k subroutines making the sound of a doorbell, telephone, sirens, bombs, whistles, ticking clock, sawing wood. I've also typed in a program to be found on page 39 which editor Dave Small says will cause sales of Apples to drop wherever it's shown. Called "Sunset", I agree it is a dramatic demonstration of graphics.

We Atari users are finding an increasing number of valuable books to help us use the Atari to its best advantage. The Creative Atari is one book which will not gather dust on shelves.

—Jim Bumpas

BRYAN'S ARCADE

Zaxxon is now out on disk and tape for the Atari 400/800. Zaxxon is available through Data-Soft Inc. for \$39.95.

Data-Soft's version of Zaxxon is probably the closest version ever released for the Atari home computer.

If you like Zaxxon in the arcade you will definitely like this game. The graphics are almost exactly like the arcade version. The city and the Zaxxon robot are both excellent demonstrations of what the Atari computer is capable of doing.

If you have both a disk drive and a tape recorder and you're going to get a copy of Zaxxon I highly recommend you get the disk version because it is much better than the cassette version. For one thing the disk version of Zaxxon has by far much better graphics than the cassette version. The cassette version does not have little bombs coming up at you all over the city like the disk version has. The sound, color, animation, graphics and arcade action are very much like the arcade version. So if you like Zaxxon you should really enjoy Zaxxon on disk or cassette by Data-Soft Inc.

— Bryan Dunn

NECROMANCER

Necromancer is a exciting wizardry game full of adventure. The object of the game is to kill the evil wizard down in the gloomy grave yard. There are two screens to the game before you enter into the grave yard.

What happens in the first screen? You are located in the middle of the page with Ogres on all sides of you trying to stomp down the army of trees you are planting. You move around your magic wand to control where you will deposit the seeds.

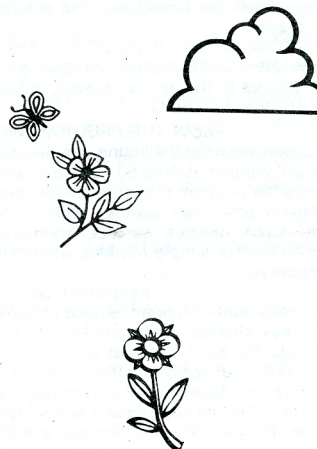
On the second screen you enter into the vaults of the unknown. Your object is to kill as many spider larvae as possible to help you in the NECROMANCER's lair. One thing making it very interesting is the Hands of Fate. They will pick you up and put you at the top of the screen. You better be quick because if a larvae turns into a spider you are almost surely doomed.

The final task you have to complete is to clear the NECROMANCER's Lair. This level takes a lot of skill and quick thinking to complete. The object is to knock down the tomb stones so his spirit can't come back to life.

The graphics on this game are super and the same goes for the music. I think this is a very exciting game and I highly recommend it to anyone.

by Synapse Software, 1(415)527-7751, 5221 Central Av., Richmond CA. 94804

review by L. J. Knoll



The Pilot's Sketch Pad

The Pilot's Sketch Pad is a program letting you use a joystick to draw pictures on your screen. You can change 'pen' size and color as you draw. You can even save your sketch on disk or tape for later recall. This is an example of the type of program ATARI PILOT handles very nicely. PILOT may not be too hot for numerical calculus or accounts payable but it handles interactive graphic functions in an easy way.

HOW TO USE THE PILOT'S SKETCH PAD

First let's take a look at how to use the program. Then, if you are interested, you can read on to see what makes it tick. When you first RUN the program you will see a graphic screen, blank except for a single flashing yellow dot. This dot is the 'pen'. As you move the joystick the pen will leave a trace. If you push the trigger the pen turns into an eraser. By pressing the 'R', 'B', and 'Y' keys you can switch between the red, blue, and yellow pens. You can also switch between the three pen sizes by using the '1', '2', and '3' keys.

The rest of the program control functions (Save, Load, Clear, Quit) are 'protected' since they cause more final actions. To use these functions you must hold down the Control 'CTRL' key and then press the desired function key. Any of these functions can be used at any time.

To CLEAR the screen - Press CTRL C.

To SAVE the screen - Press CTRL S. The program will then prompt you to enter a file name. To save on tape simply type 'T:' (don't type the "marks). To save on disk type 'D:filename'. For example, to save the sketch of a house on disk, type 'D:HOUSE.PIX'. It takes about a minute to save the screen on disk. Saving on tape takes somewhat longer. Once the screen has been saved the program returns to the normal 'sketch' mode.

To LOAD the Screen - Press CTRL L. The program will ask for the file name (see SAVE option). Following the LOAD function the program returns to the normal 'sketch' mode.

To QUIT - Press CTRL Q.

HOW THE PROGRAM WORKS

As you can see from the listing, the Pilot's Sketch Pad consists of a short 'Main' program (LINES 60 - 210) and a collection of subroutines. This keeps the program structure simple and the program size small. The program has three features which may be of some interest: keyboard input, joystick handler, and save/load functions. These features all involve simple routines which you can easily use in your own programs.

Keyboard Input

The various control functions used in the program are all activated by single key strokes. When any key is pressed the Atari operating system puts the key's 'internal code' in location 764. The program uses the @B+ (at byte) function to continually check to see when specific keys are depressed. For example, statement 70 sets +K equal to the code of the last key pressed, and statement 470 checks to see if it was the 'R' key. This is a very quick and easy way to get simple keyboard input. It avoids making the program stop and wait for input using the A: (ask) command.

If you want to use this technique in your program you may need to find the internal codes for a few characters. The following short program will display the internal code of any key pressed.

```
10 *START
20 T:@B764
30 J:*START
```

Joystick Handler

Using the joystick in PILOT can sometimes be a bit inefficient unless you understand how the ATARI joystick registers work. Each joystick has a four bit register. The bits in this register correspond to the four primary joystick positions: right, left, down, and up. The following are some examples of the register readings for different stick positions:

RLDU: 0001 - Up x 1000 - Right x 1001 - Right and Up x 0110 - Left and Down

By dividing the register value by 1, 2, 4 or 8 and using the modulo 2 (1/2) operation we can check the individual bits of the register. For example, statement 660 checks the 'R' bit and adds the pen width (+P) to the horizontal position (+X). A check is made to insure the pen stays on the screen.

Save/Load Functions

Saving and reloading the screen is really quite easy. PILOT uses a graphics-7 type display list. The ATARI operating system uses locations 88 and 89 to point to the start of graphics-7 display list (see statement 290). The display list is 3200 bytes long. The program uses the @+ (at number) function to read or write two bytes at a time using the designated file name. This technique may not be very fast (about 1 minute for disk), but it is very simple and it is 100 percent PILOT.

I hope you will enjoy the Pilot's Sketch Pad and find some of these routines useful in your own programs.

—Jim A. Carr - Corvallis, OR

TITAN LANDER

by Dale Lutz, Alberta, Canada

A long time ago in a country far, far away I saw an arcade game I wanted to write for my Atari 800. At the time I knew little of P/M graphics, priority and color registers, and nothing about machine language and collision registers. The reason I am telling you this is to give all of you beginners out there some tips and hope.

The first version of Lander I did used GRAPHICS 5, and poorly designed, blocky, BASIC-moved P/M graphics. There was no way to win. Since I knew nothing of collision registers there was no way, short of extensive coordinate checking, to detect crashes and landings.

Then I purchased the Master Memory Map from Santa Cruz, bought De Re Atari, subscribed to Creative Computing and COMPUTE!, and became an ACE member. After reading many, many articles, I completely rewrote the original lander using single line P/M resolution, and Antic mode E (see Outpost Atari, March 1982 Creative Computing). The way I did it, though, it would only work on cassette based 32-48K systems. I recently revised the program so it runs in GRAPHICS 7 on any 32K machine (24K without a disk). The final game, I'm sure you'll agree, is fairly good considering it is done completely in BASIC.

A few notes on typing the game in: Type NEW, and then enter LINE 1. It is imperative that line 1 is absolutely the first line entered, because string PM\$ must be the first variable in the variable name table. (If after you get the thing typed in, and no ship appears on the screen, try LISTing it out, typing NEW, and then ENTERing it back in). This is because we modify its string storage pointers in LINE 20025 to point to the P/M memory area. This allows us fairly fast vertical positioning without dirtying our hands with machine language. What we do is first find the start of the variable table, calculate the offset of the array table, and POKE it in the right place (for a detailed explanation, read the last few pages of Outpost Atari in the January 1982 Creative Computing). This is the same technique Mr. Brown used in Eggs.

On playing the game: you'll need a joystick plugged into port#1. Push whichever way you want thrust. The yellow bar graph is representative of your vertical velocity. Your ship will appear from out of the top left hand corner. When you land on one of the flat horizontal pads, this bar must be between the two green horizontal lines. The game requires quite a bit of skill, so be prepared to do some practice on the easy sights before attempting some of the harder pads. When the yellow bar graph turns red, you only have a little fuel left. When you're out of fuel, you can only watch as your ship plummet to its death. Should you successfully land, however, you be treated to some entertainment and, more importantly, some more fuel.

Following is a line by line breakdown of the program:

LINE # - Functions

1 - DIMs all the strings the program uses
5 - Jumps all over for program initialization
10-55 - Main program loop (get stick, move ship, check for collisions, etc.)
60-85 - Program comes here to find out if landing was legal
87-199 - Produce crash and wait for START to be pushed
200-290 - Routines for a successful landing
310 - Program goes here if you go out of the screen area
7000-7999 - Draws playfield
8000-8999 - Set up some equates
20000-20999 - Set up strings for P/M graphics
21000-21010 - Clear out P/M display memory using strings

The program itself is not the end-all. In order to keep motion close to smooth without using machine language, I have had to sacrifice constant display of fuel remaining. The entire display could be redone in Antic mode 4, keeping great resolution, lowering the memory requirement, and hastening program execution.

In closing, never give up. Remember: Even without machine language you can still write some pretty decent action games, if you know a few BASIC tricks. We all need more FREE public-domain games!

☆☆☆☆☆☆

PILOT

```

1 R:*****
2 R:** ACE NEWSLETTER **
3 R:** 3662 Vine Maple **
4 R:** Eugene, Or 97405 **
5 R:** $10 year **
6 R:** June 1983 **
7 R:*****
10 R:** PILOT'S SKETCH PAD **
20 R:
30 R:Jim A. Carr 1-May-83 Corvall
is OR
40 R:*****
50 U:*CLEAR
60 *MAIN [Start of main program]
70 C:K=0B764 [Check for key pres
sed]
80 C:0B764=255 [Reset keyboard in
put]
90 U(K=146):*CLEAR [ctrl C]
100 U(K=190):*SAVE [ctrl S]
110 U(K=128):*LOAD [ctrl L]
120 E(K=175): [ctrl Q]
130 U:*JSTICK [Track joy stick]
140 U:*GETCOLOR
150 U:*PENSIZE
160 U:*PLOT
170 GR:PEN ERASE
180 U:*PLOT
190 U(178=0):*GETCOLOR [Erase if
trig]
200 U:*PLOT
210 J:*MAIN
220 R:-----
230 *SAVE [Save or load picture b
y]
240 *LOAD [accessing display area
.]
250 T: ENTER FILE NAME
260 T: DISK CASSETTE
270 T: D:filename C:
280 A:$FILENAME
290 C:$A=256*0B89 +0B88 [display
area]
300 C:$E=$A+3200
310 *NEXTBYTE
320 WRITE(K=190):$FILENAME @ $A
330 READ (K=128):$FILENAME @ $A
340 C:$A=$A+2
350 J($A<$E):*NEXTBYTE
360 CLOSE:$FILENAME
370 U:*INSTRUCTIONS
380 E:

```

```

390 R:-----
400 *PLOT [Put dot at X,Y]
410 GR:GOTO $X,$Y [small]
420 GR(K=2):GOTO $X,$Y+1;GOTO $
X+1,$Y+1;GOTO $X+1,$Y [medium]
430 GR(K=3):TURNTO 90;GO 1;2(TUR
N -90;DRAW 2) [large]
440 E:
450 R:-----
460 *GETCOLOR [Set color to match
key]
470 C(K=40):C=1 [R=red]
480 C(K=43):C=2 [Y=yellow]
490 C(K=21):C=3 [B=blue]
500 GR(C=1):PEN RED
510 GR(C=2):PEN YELLOW
520 GR(C=3):PEN BLUE
530 E:
540 R:-----
550 *PENSIZE [Set pen size to mat
ch key]
560 C(K=31):P=1 [1=small]
570 C(K=30):P=2 [2=medium]
580 C(K=26):P=3 [3=large]
590 E:
600 R:-----
610 *JSTICK [Joystick sets XY pos
]
620 C:$J=$J0
630 C($Y< 47):$Y=$Y +(($J\2)*P)
640 C($Y>31):$Y=$Y -(($J\2)*P
)
650 C($X>79):$X=$X -(($J\4)*P
)
660 C($X< 79):$X=$X +(($J\8)*P
)
670 E:
680 R:-----
690 *CLEAR [Display key codes]
700 GR: CLEAR
710 *INSTRUCTIONS
720 T: COLOR LINE SIZE CON
TROL KEYS
730 T:R. RED 1. SMALL 5. SA
VE Q. QUIT
740 T:B. BLUE 2. MEDIUM L. LO
AD
750 T:Y. YELLOW 3. LARGE C. CL
EAR\

```

```

760 C:P=1 [Initialization]
770 C:C=2
780 C:0B764=255
790 E:
10 T: THE PILOT'S SKETCH PAD
20 T: =====
30 R:
40 R:By Jim Carr 03-May-83 Corvallis OR
50 R:
60 T:
70 T:WHAT IS IT?
80 T:
90 T:The Pilot's Sketch pad is a program
100 T:which allows you to create color
110 T:pictures on your ATARI computer.
120 T:You use the Joystick to guide
130 T:an imaginary pen which draws the
140 T:picture on your screen. You have
150 T:control of both the pen size and
160 T:the color of the ink. Completed
170 T:pictures can be saved on either
180 T:disk or cassette. Once saved a
190 T:picture can later be recalled for
200 T:display and further changes.
210 T:
220 T:The program is written in ATARI
230 T:PILOT.
240 U:*PAGE
250 T:HOW DO YOU GET STARTED?
260 T:
270 T:To get started with the Pilot's
280 T:Sketch Pad you must first plug a
290 T:joystick into port number one.
300 T:
310 T:The next step is to load the
320 T:the program. If you are using a
330 T:disk the program name is SKETCH.
340 T:When the program is loaded, type
350 T:RUN.
360 T:
370 T:The next thing you should see is a
380 T:small yellow dot flashing in the
390 T:center of your screen. If not,
400 T:maybe it would be a good day to go
410 T:outside and play in the rain.
420 U:*PAGE
430 T:HOW DO I DRAW A PICTURE?
440 T:

```

GOTO PAGE 9

GAME OF THE MONTH!!

TITAN LANDER by Dale Lutz Alberta, Canada

```

1 DIM PM$(2048),LAN$(25),LT$(19),
RT$(19),UPT$(16),BL$(30),V$(45)
2 REM TITAN LANDER REV. 17
   (GRAPHICS 7 VERSION)
3 REM BY DALE LUTZ
   REQUIRES AT LEAST 32K
WITH DOS      (24K WITHOUT)
4 REM LINE 1 MUST BE THE FIRST LI
NE      ENTERED
5 POKE 559,0:GOSUB 20000:GOSUB 80
00:GOSUB 7000:POKE 710,0:POKE 559
,62:F=200:SC=0:H5=0
6 REM *****
7 REM *   ACE NEWSLETTER   *
8 REM * 3662 Vine Maple Dr. *
9 REM * Eugene, Or 97405   *
   *       $10yr       *
   *****
10 Y=15:X=30:XT=1:YT=-0.125:POKE
53278,255:TRAP 300:POKE 711,22
20 A=5TICK(0):PM$(PY1+Y)=BL$:PM$(
PY2+Y)=BL$:PM$(PY3+Y)=BL$:IF Y<0
THEN 300
22 IF F<50 THEN POKE 711,48
23 IF F<0 THEN A=15
25 IF A=15 OR A=13 THEN SOUND 0,0
,0,0:GOTO 47
30 IF A=14 THEN YT=YT+0.125:POKE
HP3,X:PM$(PY3+Y)=UPT$:SOUND 0,121
,0,10:F=F-2:GOTO 47
33 IF A=6 THEN A=7:YT=YT+0.125:PO
KE HP3,X:PM$(PY3+Y)=UPT$:SOUND 0
,121,0,10:F=F-2:GOTO 45
34 IF A=10 THEN A=11:YT=YT+0.125:
POKE HP3,X:PM$(PY3+Y)=UPT$:SOUND
0,121,0,10:F=F-2
40 IF A=11 OR A=9 THEN XT=XT-0.06
25:POKE HP2,X+8:PM$(PY2+Y)=LT$:SO
UND 0,121,0,10:F=F-2:GOTO 47
45 IF A=7 OR A=5 THEN XT=XT+0.062
5:POKE HP1,X-8:PM$(PY1+Y)=RT$:SOU
ND 0,121,0,10:F=F-2
47 X=X+XT:Y=Y-YT:YT=YT-0.03125:PM
$(888-YT*10)=V$
50 POKE HP0,X:PM$(Y+PY0)=LAN$
55 IF PEEK(53252)<>1 THEN 20
60 IF X>68 AND X<77 AND (Y)=171 A
ND Y<=175 THEN M=1:GOTO 200

```

```

65 IF X>95 AND X<102 AND (Y)=70 A
ND Y<=75 THEN M=2:GOTO 200
70 IF X>130 AND X<135 AND (Y)=90
AND Y<=96 THEN M=3:GOTO 200
75 IF X>147 AND X<153 AND (Y)=125
.5 AND Y<=133 THEN M=4:GOTO 200
80 IF X>167 AND X<184 AND (Y)=170
AND Y<=175 THEN M=5:GOTO 200
85 IF X>181 AND X<186 AND (Y)=70
AND Y<=75 THEN M=6:GOTO 200
87 GOSUB 21000
90 PM$(Y+PY0)=BL$:POKE 704,14:FOR
D=1 TO 14:PM$(Y+D+PY0)=CHR$(RND(
0)*247+D):SOUND 0,100,0,D:NEXT D
100 FOR D=1 TO 14:PM$(Y+D+PY0)=CH
R$(0):SOUND 0,D*16,0,14-D:NEXT D:
SOUND 0,0,0,0
110 IF F<=0 THEN ? "YOU RAN HER O
UT OF FUEL!!":GOTO 114
112 ? "CRASHED!"
114 ? "YOUR SCORE WAS ";SC: ? "HIG
HSCORE ";H5: ? "PUSH START FOR A N
EW GAME";:REM START IN INVERSE
115 IF PEEK(53279)<>6 THEN 115
120 SC=0:F=200:GOTO 320
200 ? "LANDED"
210 IF YT<0 AND YT>-1.1 THEN ON M
GOTO 220,230,240,250,260,270
215 GOTO 90:REM CRASH
220 RESTORE 287:F=F+100:GOTO 280
230 F=F+125:RESTORE 288:GOTO 280
240 PM$(PY0+Y)=LAN$:PM$(PY3+Y)=UP
T$:Y=Y-1:IF Y<-10 THEN F=F+125:GO
TO 283
245 SOUND 0,121,0,10:GOTO 240
250 F=F+200:RESTORE 289:GOTO 280
260 F=F+150:RESTORE 290:GOTO 280
270 RESTORE 291:F=F+100:GOTO 280
280 READ P,X:IF P>0 THEN SOUND 0
,P,10,10:SOUND 1,P-1,10,10:FOR D=1
TO X:NEXT D:SOUND 0,0,0,0:SOUND
1,0,0,0:GOTO 280
283 SC=SC+F:IF SC>H5 THEN H5=SC
285 SOUND 0,0,0,0:SOUND 1,0,0,0: ?
CHR$(28);"FUEL: ";F: ? "SCORE:
";SC: ? "HIGHSCORE: ";H5:;FOR D=1
TO 800:NEXT D
286 GOTO 320
287 DATA 144,55,128,55,121,55,128
,55,144,100,121,100,182,150,1,55,
162,75,91,125,96,75,95,125,-1,0

```

```

288 DATA 182,100,162,50,153,50,18
2,75,1,150,182,100,162,50,153,50,
182,75,1,15,128,150,-1,0
289 DATA 130,30,100,30,80,30,65,7
0,80,30,65,100,-1,0
290 DATA 96,75,85,75,108,75,216,7
5,144,200,-1,0
291 DATA 121,40,108,40,96,40,108,
40,121,40,108,40,96,78,121,90,121
,170,-1,-1
300 SOUND 0,0,0,0: ? CHR$(125);"OU
T OF SKY!!"
310 FOR D=1 TO 200:NEXT D: ? "PUSH
START FOR A NEW GAME":GOTO 115:R
EM START IN INVERSE
320 PM$(1)=CHR$(0):PM$(2048)=CHR$
(0):PM$(2)=PM$:GOSUB 20040:GOTO 1
0
7000 RESTORE 7000:REM DRAW PLAYFI
ELD
7002 GRAPHICS 7:POKE 752,1:POKE 5
59,0
7005 COLOR 1:PLOT 0,32/2:DRAWTO 1
,39/2
7010 TRAP 7020:READ X,Y:DRAWTO X,
Y/2+0.5:GOTO 7010
7020 DATA 2,48,4,56,8,60,12,64,18
,72,20,76,18,88,16,104,22,112,25,
120,27,131,22,144,16
7025 DATA 152,14,157,36,157,42,14
4,44,132,46,120,50,108
7030 DATA 48,88,50,80,46,64,48,56
,60,56,78,48,80,74,84,76,94,76,95
,46,100,54,105,73,105,94
7035 DATA 97,96,95,102,101,112,11
1,112,117,136,112,156,121,157,144
7040 DATA 157,143,144,134,122,130
,98,136,72,140,67,134,56,144,56,1
50,45,148,37,151,32,159,36,END
7050 COLOR 2:PLOT 149,45:DRAWTO 1
59,45:PLOT 149,50:DRAWTO 159,50
7999 RETURN
8000 PY1=1284:PY0=1024:PY2=1539:P
Y3=1805:HP0=53248:HP1=53249:HP2=5
3250:HP3=53251
8010 POKE 53252,199:POKE 53253,19
9:POKE 53254,199:POKE 53255,199
8999 RETURN
20000 REM SET UP THE STRINGS TO H
ANDLE PM MOVING

```



```

20010 POKE 704,12:POKE 705,54:I=P
EEK(106)-32:POKE 54279,I:POKE 532
77,3:AT=PEEK(140)+256*PEEK(141):P
OKE 623,24
20020 VT=PEEK(134)+256*PEEK(135):
POKE 706,54:POKE 707,56:POKE 708,
10:POKE 53260,3:POKE 711,22
20025 OFF5=I*256-AT:HI=INT(OFF5/2
56):LO=OFF5-HI*256:POKE VT+2,LO:P
OKE VT+3,HI
20030 PM$(1)=CHR$(0):PM$(2048)=CH
R$(0):PM$(2)=PM$
20032 RESTORE 20085:FOR A=1 TO 24
:READ I:LAN$(A,A)=CHR$(I):NEXT A
20034 RESTORE 20090:FOR A=1 TO 19
:READ I:RT$(A,A)=CHR$(I):NEXT A
20035 RESTORE 20095:FOR A=1 TO 19
:READ I:LT$(A,A)=CHR$(I):NEXT A
20040 V$(1)="?":V$(43)="?":V$(2)=
V$:FOR A=1 TO 5:V$(A,A)=CHR$(0):N
EXT A
20042 RESTORE 20100:FOR A=1 TO 15
:READ I:UPT$(A,A)=CHR$(I):NEXT A
20045 BL$(1)=CHR$(0):BL$(30)=CHR$
(0):BL$(2)=BL$
20050 PM$(916)=V$:POKE 752,1
20060 ? CHR$(125);CHR$(30);CHR$(3
0);CHR$(27);CHR$(28);CHR$(27);CHR
$(29);
20070 RETURN
20085 DATA 0,0,0,0,0,24,36,36,66,
66,66,129,129,129,129,66,36,90,12
9,0,0,0,0,0
20090 DATA 0,0,0,0,0,8,4,2,9,7,9,
2,4,8,0,0,0,0,0
20095 DATA 0,0,0,0,0,16,32,64,224
,144,224,64,32,16,0,0,0,0,0
20100 DATA 0,0,0,0,0,24,36,40,153
,165,0,0,0,0,0
21000 PM$(1)=CHR$(0):PM$(2048)=CH
R$(0):PM$(2)=PM$
21010 PM$(916)=V$:POKE 752,1:RETU
RN

```

YOUR PROGRAM,

REVIEW,

ARTICLE, OR

COMPUTER ART

COULD BE HERE!!!

JUNE MEETING

TUESDAY (!)

June 7, 1983 at 7.30 pm. In the Cafeteria of South Eugene HS. Entrance is through the rear. Look for the signs!

This meeting will be video-taped by a crew from ATARI. The tape will be used to create a presentation to show to other users' groups as a part of the Atari Users' Group support program.

OCKERS

Stan Ockers Listings

```

1 REM *****
2 REM ** Ace Newsletter **
3 REM ** 3662 Vine Maple **
4 REM ** Eugene, Or 97405 **
5 REM ** $10 yr **
6 REM ** June 1983 **
7 REM *****
10 REM *****
20 REM ** Scrolling City **
25 REM ** Background **
30 REM ** Stan Ockers **
40 REM *****
90 GRAPHICS 7:RESTORE 92:FOR J=70
8 TO 712:READ A:POKE J,A:NEXT J
92 DATA 32,14,86,84,8
100 GOSUB 1010:GOSUB 2000:GOSUB 2
200:GOSUB 2300:GOSUB 2100:POKE 75
6,CSPAGE
115 A=USR(1580)
120 GOTO 120
1000 REM * change character set *
1010 DIM MCS$(42):RESTORE 1020:FO
R J=1 TO 42:READ A:MCS$(J,J)=CHR$
(A):NEXT J
1020 DATA 104,169,0,133,203,133,2
05,169,224,133,204,165,106,56,233
,21,133,106,24
1030 DATA 105,1,133,206,162,4,160
,0,177,203,145,205,200,208,249,23
0,204,230,206,202,208,242,96
1040 A=USR(ADR(MCS$)):CSPAGE=PEEK
(106)+1:C5=256*CSPAGE:SCRST=C5+10
24
1050 RESTORE 1060:FOR J=C5+776 TO
C5+968:READ A:POKE J,A:NEXT J:RE
TURN
1060 DATA 255,219,219,219,219,219
,219,255,150,150,150,170,150,150,
150,150,85,125,125,125,125,125,12
5,85
1070 DATA 85,105,105,105,105,105,
105,85,255,215,215,125,125,215,21
5,255,170,251,170,191,170,251,170
,191
1080 DATA 169,165,101,101,170,105
,89,90,0,0,0,0,0,255,255,235,23
5,255,235,235,255,235,235

```

```

1090 DATA 85,101,101,101,101,101,
101,85,255,215,255,215,255,215,25
5,215,219,182,109,219,182,109,219
,118
1999 REM * Create New Display Lis
t *
2000 FOR J=1539 TO 1572 STEP 3:PO
KE J,85:NEXT J
2002 DIM BHI(12),BLO(12)
2003 FOR J=1 TO 12:B=5CRST+256*J-
46:BHI(J)=INT(B/256):BLO(J)=B-256
*BHI(J):NEXT J
2005 FOR J=1536 TO 1538:POKE J,11
2:NEXT J:K=1540:FOR J=1 TO 12:POK
E K,BLO(J):POKE K+1,BHI(J):K=K+3:
NEXT J
2010 POKE 1575,65:POKE 1576,0:POK
E 1577,6:POKE 560,0:POKE 561,6:RE
TURN
2099 REM * Init. Page 6 Values *
2100 RESTORE 2102:FOR J=1664 TO 1
667:READ A:POKE J,A:NEXT J
2102 DATA 1,1,2,0
2104 REM * VBI Scrolling Routine
*
2105 DIM VBI$(159):RESTORE 2108:F
OR J=1 TO 159:READ A:VBI$(J,J)=CH
R$(A):NEXT J
2107 A=ADR(VBI$):HI=INT(A/256):LO
=A-256*HI:POKE 1582,LO:POKE 1584,
HI:RETURN
2108 DATA 173,128,6,240,93
2109 DATA 169,0,141,129,6,173,120
,2,72,41,4,208,3,238,129,6,104,41
,8,208,3,206,129,6
2110 DATA 173,129,6,16,9,173,130,
6,201,1,144,57,176,7,173,130,6,20
1,216,176,48
2111 DATA 173,129,6,240,43,16,15,
206,131,6,16,30,169,3,141,131,6,1
41,4,212,208,29,238,131,6,173,131
,6
2112 DATA 201,4,144,10,169,0,141,
131,6,141,4,212,240,9,173,131,6,1
41,4,212,24,144,55
2120 DATA 216,162,0,173,129,6,48,
16,189,4,6,56
2130 DATA 233,1,157,4,6,176,19,22
2,5,6,144,14
2140 DATA 189,4,6,24,105,1,157,4,
6,144,3,254,5,6,232,232,232,224,3
7,144,214

```

```

2145 DATA 24,173,130,6,109,129,6,
141,130,6,76,98,228
2199 REM * Insert VBI Routine *
2200 RESTORE 2210:FOR J=1580 TO 1
589:READ A:POKE J,A:NEXT J:RETURN
2210 DATA 104,160,132,162,6,169,7
,76,92,228
2299 REM * Draw Buildings *
2300 RESTORE 2360:B=5CRST
2310 READ WAL,ROW,COL,WID:IF WAL=
0 THEN 2400
2320 FOR J=ROW TO 12:START=B+256*
J+COL
2330 FOR K=1 TO WID:POKE START+K,
WAL:NEXT K
2340 NEXT J:GOTO 2310
2360 DATA 102,1,0,2,103,11,2,6,97
,5,8,8,98,8,14,6,102,11,20,6,105,
3,30,8,100,5,26,6,101,8,34,4,99,6
,52,8
2370 DATA 107,4,60,6,108,7,64,6,9
7,3,82,8,103,10,68,18,106,7,90,4,
107,5,94,6,100,8,106,8,105,10,96,
12
2380 DATA 98,6,114,6,102,12,120,6
,106,4,126,8,101,8,134,4,230,11,1
48,64,99,8,150,12,233,9,164,6,97,
4,170,6
2390 DATA 100,9,178,8,99,9,194,6,
98,10,188,8,107,10,202,8,105,7,21
2,16,97,10,226,8,106,8,240,14,0,0
,0,0
2400 RESTORE 2460
2410 READ ROW,COL,WID:IF ROW=0 TH
EN 2500
2420 START=5CRST+256*ROW+COL+1:FO
R J=START TO START+WID-1:POKE J,2
32:NEXT J
2430 GOTO 2410
2460 DATA 10,2,6,10,20,6,10,38,6,
9,76,6,76,9,6,4,94,6,11,120,6,10,
138,6,8,164,6,8,194,6,11,234,6,0,
0,0
2500 RESTORE 2560
2510 READ ROW,COL,WID:IF ROW=0 TH
EN RETURN
2520 START=5CRST+256*ROW+COL+1:FO
R J=START TO START+WID-1:POKE J,1
01:NEXT J
2530 GOTO 2510
2560 DATA 8,38,8,11,38,16,6,76,6,
8,138,6,11,138,10,0,0,0

```


PILOT'S SKETCHPAD con't

```

10 ; *****
20 ; ** HORIZONTAL SCROLL ROUTINE **
30 ; **      5.0.  5-83      **
40 ; *****

0000 50      *= $0680
0680 60 HENAB *= *+1      ; (1664) zero, not enabled
0681 70 DIR   *= *+1      ; (1665) direction flag
0682 80 POS   *= *+1      ; (1666) position of screen center
0683 90 HSHAD *= *+1      ; (1667) fine scroll shadow
0025 0100 LASTX = $25      ; limit to scrolled lines
0001 0110 LOLIM = 1        ; lowest screen position
0008 0120 HILIM = 216      ; highest screen position
0278 0130 STICK0 = $278    ; joystick register
0604 0140 LOBYTE = $0604   ; first line screen data
D404 0150 HSCROL = $D404    ; fine scroll register
E462 0160 XITVBI = $E462    ; exit vertical blank
0684 0170      *= $0000

0000 AD8006 0180      LDA HENAB      ; Is horiz. scroll enabled?
0003 F05D 0190      BEQ OUT1        ; No
0005 A900 0200      LDA #0          ; Update direction of scroll
0007 8D8106 0210      STA DIR
000A AD7802 0220      LDA STICK0     ; Using joystick zero
000D 4B 0230      PHA              ; Save for later
J00E 2904 0240      AND #4          ; Bit two zero?
0010 D003 0250      BNE OVER1       ; No, not player left
0012 EE8106 0260      INC DIR        ; DIR=1 bkgd moves right
0015 68 0270 OVER1  PLA            ; Recover stick value
0016 2908 0280      AND #8          ; Bit three zero?
0018 D003 0290      BNE OVER2       ; No, not player right
001A CE8106 0300      DEC DIR        ; DIR=255 bkgd moves left
001D AD8106 0310 OVER2  LDA DIR      ; Check for limits
0020 1009 0320      BPL RTCK        ; Not left bkgd
0022 AD8206 0330      LDA POS        ; bkgd further left?
0025 C901 0340      CMP #LOLIM
0027 9039 0350      BCC OUT1        ; No more room
0029 B007 0360      BCS FSCROL      ; O.K. skip right check
002B AD8206 0370 RTCK  LDA POS      ; bkgd further right?
002E C9D8 0380      CMP #HILIM
0030 B030 0390      BCS OUT1        ; No more room
0032 AD8106 0400 FSCROL  LDA DIR      ; Fine scrolling
0035 F02B 0410      BEQ OUT1        ; Forget it
0037 100F 0420      BPL BKGRIT      ; Must be right scroll
0039 CE8306 0430      DEC HSHAD      ; One fine scroll left
003C 101E 0440      BPL NOFULL      ; Not ready for full char.
003E A903 0450      LDA #$03        ; Full scroll, reset fine scroll
0040 8D8306 0460      STA HSHAD
0043 8D04D4 0470      STA HSCROL
0046 D01D 0480      BNE FULL        ; Forced jump
004B EE8306 0490 BKGRIT  INC HSHAD    ; One fine scroll right
004B AD8306 0500      LDA HSHAD      ; Ready for full char. scroll?
004E C904 0510      CMP #$04

450 T:Try moving the joystick control in
460 T:all directions. You will see that
470 T:the little dot moves in the direction
480 T:you push the stick. Also, as the
490 T:dot moves it leaves a trail.
500 T:
510 T:Next press the "2" key on the
520 T:keyboard. The little yellow dot
530 T:will change to a medium sized dot.
540 T:This dot will leave a wider trail
550 T:as you control its position with
560 T:the joystick. Next press the "3"
570 T:key. The dot will change to a
580 T:large dot and leave a wide trail.
590 T:To get back to a small or medium
600 T:size dot just press "1" or "2".
610 U:*PAGE
620 T:HOW DO I ERASE?
630 T:
640 T:By now you probably have quite a mess
650 T:on your screen. Try pressing the
660 T:joystick trigger. The dot will
670 T:continue to flash, but as you move
680 T:the dot around with the joystick
690 T:instead of drawing, it erases.
700 T:
710 T:HOW DO I CHANGE COLOR?
720 T:
730 T:By now you may also be getting a bit
740 T:tired of yellow. Try pressing the
750 T:"R" key. The dot will change to
760 T:red. The "B" key will change the
770 T:dot to blue and the "Y" key will
780 T:change back to yellow. You can use
790 T:these three keys at any time to
800 T:switch between yellow, red, and blue.
810 T:
820 T:Now is is up to you to draw a picture.
830 U:*PAGE
840 T:HOW DO I SAVE A PICTURE?
850 T:
860 T:To save your picture, simply press
870 T:"CTRL 5". That is, while holding the
880 T:"CTRL" key, press the "5" key. The
890 T:program will respond by asking for a
900 T:file name. To save your picture
910 T:on disk enter something like
920 T:"D:HOUSE.PIX" (if your picture is
930 T:a house). To save your picture on
940 T:cassette, enter "C:". The computer
950 T:will beep twice to remind you to
960 T:set up the recorder to record.

```


OCKERS'

PILOT'S SKETCHPAD con't.

```

0050 900A 0520      BCC NOFULL      ; No, not yet
0052 A900 0530      LDA #000        ; Reset fine scroll
0054 8D8306 0540     STA HSHAD
0057 8D04D4 0550     STA HSCROL
005A F009 0560      BEQ FULL        ; Forced jump
005C AD8306 0570 NOFULL LDA HSHAD    ; Update fine scroll register
005F 8D04D4 0580     STA HSCROL
0062 18 0590 OUT1   CLC              ; Forced jump to exit
0063 9037 0600      BCC OUT
0065 D8 0610 FULL   CLD              ; Full character scroll
0066 A200 0620      LDX #000        ; Index on first line
0068 AD8106 0630 LOOP LDA DIR        ; Must check direction
006B 3010 0640      BMI LEFT        ; Bkgd moves left
006D BD0406 0650     LDA LOBYTE,X    ; Bkgd right, pointer back one
0070 38 0660      SEC
0071 E901 0670      SBC #01
0073 9D0406 0680     STA LOBYTE,X
0076 B013 0690      BCS NEXT        ; No page crossing
0078 DE0506 0700     DEC LOBYTE+1,X ; Down a page
007B 900E 0710      BCC NEXT        ; Forced jump
007D BD0406 0720 LEFT LDA LOBYTE,X  ; Bkgd left, pointer up one
0080 18 0730      CLC
0081 6901 0740      ADC #01
0083 9D0406 0750     STA LOBYTE,X
0086 9003 0760      BCC NEXT        ; No page crossing
0088 FE0506 0770     INC LOBYTE+1,X ; Up a page
008B E8 0780 NEXT   INX              ; Point to next line
008C E8 0790      INX
008D E8 0800      INX
008E E025 0810      CPX #LASTX      ; Is this the last line?
0090 90D6 0820      BCC LOOP        ; No do another
0092 18 0830      CLC              ; Update screen position
0093 AD8206 0840     LDA POS
0096 6D8106 0850     ADC DIR        ; Using direction value
0099 8D8206 0860     STA POS
009C 4C62E4 0870 OUT JMP XITVBI     ; Back to Atari VBI

```

```

970 T:
980 T:It takes about a minute to save
990 T:a picture on disk. It takes
1000 T:several minutes to save on tape.
1010 T:When the picture has been saved,
1020 T:the program returns to the flashing
1030 T:dot.
1040 U:*PAGE
1050 T:HOW DO I RELOAD A SAVED PICTURE?
1060 T:
1070 T:To load a saved picture type
1080 T:"CTRL L". The program will again
1090 T:ask for the file name.
1100 T:If the picture was saved on disk type
1110 T:"D:filename" where "filename" is
1120 T:is the name you used when you saved
1130 T:the picture. If the picture was
1140 T:saved on tape, type "C:".
1150 T>Loading pictures is a bit faster
1160 T:saving, but not much.
1170 T:
1180 T:If you want to start a new picture
1190 T:you can clear the entire screen by
1200 T:pressing "CTRL C".
1210 T:
1220 T:WHAT DO I DO WHEN I AM ALL DONE?
1230 T:
1240 T:When you are ready to quit press
1250 T:"CTRL Q".
1260 E:[Thats all folks]
1270 *PAGE [Page control routine]
1280 T:
1290 T:Press RETURN to continue.\
1300 A:
1310 GR:QUIT
1320 E:

```

0680 HENAB	0681 DIR	0682 POS	0683 HSHAD
=0025 LASTX	=0001 LOLIM	=0008 HILIM	=0278 STICK0
=0604 LOBYTE	=D404 HSCROL	=E462 XITVBI	0062 OUT1
0015 OVER1	001D OVER2	002B RTCK	0032 FSCROL
0048 BKGR	005C NOFULL	0065 FULL	009C OUT
0068 LOOP	007D LEFT	008B NEXT	

PROGRAMS BY SHANE ROLIN

```

10000 REM -----
10010 REM --
10020 REM -- BLINKING CURSOR --
10030 REM --
10040 REM -- SUBROUTINE --
10050 REM --
10060 REM --
10070 REM -----
10080 REM --
10090 REM --TO LOAD MACHINE LANG.
10100 REM --SIMPLY HAVE A GOSUB
10110 REM --30000 IN YOUR PROGRAM

10120 REM --
10130 REM -----
10140 REM --
10150 REM --TO BLINK-BLINK-BLINK:
10160 REM --TO START THE CURSOR
10170 REM --BLINKING, SIMPLY PUT
10180 REM --THE FOLLOWING STATE-
10190 REM --MENT IN YOUR PROGRAM:
10200 REM --'CURSOR=USR(1536)'
10210 REM --AND THE CURSOR WILL
10220 REM --BLINK-BLINK-BLINK!
10230 REM --
10240 REM -----
30000 RESTORE 30040
30010 FOR I=1536 TO 1567:READ A
30020 POKE I,A:NEXT I
30030 RETURN
30040 DATA 104,162,6,160,11,169,6
,32
30050 DATA 92,228,96,165,20,110,2
43,2
30060 DATA 110,243,2,106,106,106,
106
30070 DATA 46,243,2,46,243,2,76,9
5,228

```

```

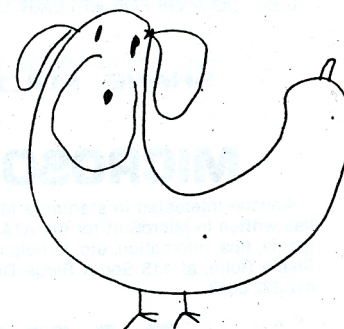
100 REM *****
110 REM **
120 REM ** RAMMER **
130 REM ** RAM TESTER **
140 REM ** by Shane Rolin **
150 REM ** VERSION 2.0 **
160 REM **
170 REM *****
1000 POKE 710,3
1010 Q1=1:Q2=Q1+Q1:Q255=255:Q256=
Q255+Q1
1020 FOR QA=1536 TO 1576:READ QB:
POKE QA,QB:NEXT QA
1030 DATA 104,169,0,160,0,24,145,
203,209,203,208,18,105,1,208,246,
200,208,242,230,204,166,204,228,2
06
1040 DATA 208,234,133,208,96,133,
207,177,203,133,209,169,1,133,208
,96
1050 QA=PEEK(15)*Q256:QB=PEEK(742
)*Q256:IF PEEK(14)<>Q0 THEN QA=QA
+Q256
1060 ? CHR$(125):POSITION 15,5:?
"RAMMER":POSITION 13,6:? "RAM TES
TER"
1070 ? :? "MEMORY BOUNDS ARE"
1080 ? "LOW=";QA;" "; "HIGH=";QB
1090 ? "ENTER LOW ADDRESS(RETURN)
":? "AND ENTER HIGH ADDRESS(RETUR
N)"
1100 ? "THE ADDRESSES MUST BE AT
LEAST 257 ":? "BYTES APART.":?
1110 TRAP 1110:? "LOW=";:INPUT Q1
Q1:IF Q1<QA OR Q1>QB THEN 1110
1120 TRAP 1120:? "HIGH=";:INPUT Q
HI:IF QHI>QB OR QHI<QA OR QHI-Q1O
<Q256 THEN 1120
1130 POKE 752,1:POKE 205,Q0:POKE
206,INT(QHI/Q256)
1140 TRAP 32767:POKE 203,Q0:POKE
204,INT(Q1O/Q256)
1150 POKE 764,Q255
1160 POKE 764,Q255:QC=USR(1536)
1170 IF PEEK(208)=Q0 THEN 1250
1180 ? " BAD LOCATION AT ";(PEEK(
203)+PEEK(204)*Q256);" EXP=";PEEK
(207);" ACT=";PEEK(209)
1190 ? CHR$(253)

```

```

1200 IF PEEK(203)=Q255 THEN POKE
204,(PEEK(204)+Q1):POKE 203,Q0:GO
TO 1220
1210 POKE 203,(PEEK(203)+Q1)
1220 POKE 764,Q255:POKE 559,34
1230 IF PEEK(764)=Q255 THEN 1230
1240 GOTO 1160
1250 QD=QD+Q1:? " SUCESSFULL PASS
NUMBER ";QD:? CHR$(253)
1260 FOR QE=Q1 TO 5:NEXT QE:SOUND
Q0,Q0,Q0,Q0
1270 IF PEEK(764)<>Q255 THEN 1290
1280 GOTO 1140
1290 POKE 764,Q255
1300 IF PEEK(764)=Q255 THEN 1300
1310 GOTO 1140

```



Hot Off The Press
An Addition To Benioff At
Large

This information was confirmed
just before printing.....

1)The all new ATARI 1400XL.
This corected version of the
1200XL will replace the 1200XL
in the line. It looks just
like a 1200XL, but has the
corrected bugs.

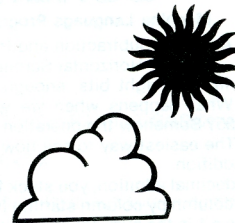
2)The all new ATARI 1400XLD.
A 1400 with two disk drives
built in above the computer.

3)A new disk drive (i.e. see
the Tandem) is only 2" tall.
It was described as being
similar to the looks of the
new Rana drive.

4)A new graphics tablet called
the "Touch Tablet". Use your
finger, pen, pencil, to create
pictures that go directly to
the screen.

.....All of this is hot
information to be released at
CES! I haven't been wrong
yet, so.....

See You in July.
Marc



Next Issue

The next issue is our
July/August summer issue, so
please get your articles in if
you don't want to wait until
September!! Pictures we can
use for the front cover,
especially made on the GTIA
Sketchpad, most appreciated.



OCKERS

Machine Language Programming #7

Addition, Subtraction and Page Crossing Horizontal Scrolling

One byte will hold eight bits, enough to form the numbers zero through 255. What happens when we want to deal with numbers greater than 255? Somehow the operation must be transferred over to another byte. The easiest way to see how this works is to look at the operation of addition.

Just like in decimal addition, you stack two numbers on top of each other and add column by column starting from the right. A 1 and a 0 (or a 0 and 1) gives a 1 in the sum column while 0 and 0 gives a zero. 1 and 1 gives a zero with a 1 CARRY. you have to remember to add the carry in on the next column to the left. See if you can follow this addition:

```
0110101 (decimal 53)
+ 0010110 (decimal 22)
-----
```

```
1001011 (decimal 75)
```

What happens if there is a carry on the leftmost column? try adding 1 to 255:

```
1111111 (decimal 255)
+ 0000001 (decimal 1)
-----
```

```
0000000 (decimal 0, carry is set)
```

The result is zero but a bit in a special location called CARRY is set to indicate the sum exceeds 255 and it is going to take two bytes to contain the sum.

To address one of the more than 64,000 addresses available in the Atari we need two bytes, a LO byte and a HI byte. To add a two byte number to an address we first add the two LO bytes, putting the result in the LO byte of the sum. The HI bytes then are added along with carry and the result put in the HI byte of the sum. Actually the only instruction for addition is ADC (Add with Carry), so all addition operations should be preceded by a CLC (Clear Carry) so the condition of carry is known.

If the number we are adding to our two byte number is 255 or less, we can simplify things by simply incrementing the high order byte rather than adding 0 + carry + HI order byte. Here is the code:

```
CLC ; Prepare for addition
LDA LOBYTE ; All done in the accumulator
ADC #NUMBER ; Number is 255 or less
BCC OVER1 ; If carry clear no overflow
INC HIBYTE ; Must have exceeded LO byte
OVER1 ... ; Rest of code
```

There are a number of addressing modes for ADC but all operate with the accumulator and the result is placed in the accumulator. Where the number comes from depends on the addressing mode.

In subtraction a digit often has to be borrowed from the column to the left. Likewise, when subtracting multiple byte numbers, borrowing can take place between bytes. Keeping track of borrows is again a function for the carry bit. The rule for subtraction is: always set carry before the subtraction. If carry is clear after subtracting two LO byte numbers, you know that a borrow needs to take place in the HI byte subtraction. The operation is SBC (Subtract with borrow).

Again for numbers 255 or less we simplify:

```
SEC ; Prepare for subtraction
LDA LOBYTE
SBC #NUMBER
BCC OVER2 ; No borrow necessary
DEC HIBYTE ; Down a page
OVER2 ... ; Rest of code
```

Dealing with two byte addresses brings about the idea of 'PAGE'. The LO byte can hold 256 different values (0-255) and in addressing every 256 bytes constitutes a PAGE. If we increment to LO byte by one after reaching 255, it 'rolls over' to zero and the high byte should be incremented up once (1 page). In subtraction the number 0 rolls over to 255. The horizontal scroll routine listed uses the ideas just presented. It is fairly complicated so some explanation is in order. In the Atari a separate processor (called Antic) handles the display of the screen. It has its own program (called the display list) which tells it what to do. Imagine 12 lines of text in mode 2 on the screen. Each line has one instruction in the display list and for at least the first, this instruction should indicate where to get the data for the screen. Remaining lines could just continue to pick up data where the previous one left off.

Instructions which include the address of where to get data are called LMS (Load Memory Scan) instructions. LMS instructions are made of three bytes. The first indicates the graphics mode and other options while the second and third form the LO and HI bytes of a pointer to where the screen data for that line is to begin.

For horizontal scrolling we need quite a bit of data (more than one screen worth). A separate string of data, enough for many screens, is provided for each line. There is an individual LMS instruction for each line. The LMS instructions are manipulated to move the screen through the data as shown below:

SCREEN

```
.....XXXXXXXXXXXXX...
.....XXXXXXXXXXXXX...
.....XXXXXXXXXXXXX...
.....XXXXXXXXXXXXX...
.....XXXXXXXXXXXXX...
```

(LMS instructions indicate where data for screen is to begin)

Code which handles the scrolling can be seen in lines 560-790 of the assembly listing. Each LMS Pointer is either incremented or decremented depending on the desired direction of movement (DIR). The use of the ideas discussed earlier can be easily seen.

In the Basic program to follow, Mode 5 rather than graphics 2 is used and a new display list is poked into page 6 as follows:

```
0600 1536 112 (decimal) ; 8 blank raster lines
0601 1537 112 ; 8 more
0602 1538 112
0603 1539 85 LO1 HI1 ; LMS instr. 1st line
0606 1542 85 LO2 HI2 ; LMS instr. 2nd line
10 more LMS instr. as above
0627 1575 65 0 6 ; Jump back to beginning
```

If updating the LMS byte was the only change involved, the scrolling would be fairly coarse, jumping one character at a time. Fortunately Atari provides fine scrolling where the movement is only part of a character width at a time. Lines 370-570 handle fine scrolling. The number in register HSCROL determines how much of a character shift is involved. In mode 5 there are 4 pixels horizontally across a character. To get the background to go left, the number in HSCROL is decremented from 3 to zero and then reset to 3 along with incrementing all LMS Pointers. Notice that when scrolling is active some movement takes place during each VBI, coarse scrolling taking place only every fourth time through. This is as fast as we can scroll without skipping some of the fine scrolling or leaving it out altogether.

It is often necessary to keep track of just what portion of the data is on the screen at any one time. POS is used for this purpose and also to provide limits to motion. Lines 280-360 handle this chore. The direction flag DIR is either +1, -1 or 0 for background right, background left or no movement respectively. DIR is added to POS when a full scroll is performed.

I've used this routine to form the background for a game. The Basic listing is called 'Scrolling City Background'. I haven't decided yet if the game has enough merit to finish, but it will give you a good idea of the routine's potential. Various characters are re-defined to form the buildings. The character set moving routine from 'Wildwest' was used except changed so RAMTOP is lowered 21 pages making room for: 1) player missile graphics - single line resolution 2) 3 K of screen data for 12 lines of horizontal scrolling 3) a new character set with some characters revised. The 3K of screen data actually overlaps 1K into an unused portion of the PM area. Before altering RAMTOP, a GR. 23 command clears the top 4K of memory.

It takes quite a bit of data to fill our 3K of screen data. I don't think most people enjoy typing all that in so Lines 2300-2900 draw the rectangular buildings. The process could be speeded up considerably by coding this in machine language. You may wish to try it.

—Stan Ockers



GTIA SKETCHPAD

MODIFICATIONS by Scott Berfield
644 W.Surf, Chicago, IL 60657

Hi there. Some of you may have typed in a program of mine called "GTIA Sketchpad". Well I have since made a few modifications which act to make it a little more friendly, as well as another version of the screendump from the first version. I also have a very simple routine to allow you to load a screen created by "GTIA S." into your own programs.

The change applies only to cassette users out there. When I wrote the save/load routines in SKETCHPAD I didn't know about the possibility of recording a file with short inter-record gaps on cassette. By formatting a file with the short gaps you shorten load/save time by almost 30-40%. To implement this change the following lines:

```
1080 OPEN #N1,4,128,FILE$
1150 OPEN #N1,4,128,FILE$
Delete line 1160
```

This will only affect cassette users, the disk drive will ignore the change.

By making this change you lose the ability to save the color registers, but I think the speed increase justifies the loss.

Screens recorded by the first version of this program will not be compatible with these changes. The easiest remedy for this is to change only the save routine; load a previously saved screen; save it; and then change the load routine.

In a drawing I was doing the other day, I had to plot many, many points and I found it tiresome to keep using the keyboard. To change this as well as the LINE and FILL functions to allow for continuous use (until you hit ESC) make the following changes & additions:

```
210 ?CHR$(125); " Place the cursor within the outline and press
trigger(ESC to exit)."
214 TRAP 40
215 IF PEEK(KEY) = 28 THEN POKE KEY,NOKEY:GOTO 40
340 YP = YP + 1:LOCATE XP,YP,COLR:IF COLR = C THEN GOTO 210
360 TRAP 40000:GOTO 40
580 ?CHR$(125); " Position cursor and press trigger to plot points(ESC
to exit)."
85 IF PEEK(KEY) = 28 THEN POKE KEY,NOKEY:GOTO 40
610 GOTO 580 x 620 ?CHR$(125); " Position cursor and press trigger to
plot the starting point(ESC to exit)."
640 ?CHR$(125); " Now position cursor at endpoint and press
trigger(ESC to exit)."
650 GOSUB JOYSTICK:PLOT CXP,CYP:DRAWTO XP,YP
655 IF PEEK(KEY) = 28 THEN POKE KEY,NOKEY:GOTO 40
656 GOTO 620
```

These changes should make things go a little more smoothly for complicated drawings.

The new screendump is quite similar to the routine used in the previous version, and I include coding to allow the choice of which you prefer to use. There are two main differences in this version: it produces a randomized pattern, and it is one of the slowest screendumps in history.

The effect produced is rather more pleasing on smoothly rounded surfaces than is the more patterned gray scale of the other version. When I say it's slow, believe me! I recently got groceries while a screen printed. I got back before it finished. For what it's worth, here it is:

```
185 IF OPT = 57 THEN GOTO 32000
32000 ?CHR$(125); "RANDOM OR PATTERNED?(ESC TO EXIT)"
32010 OPEN #3,4,0,"K":GET #3,A:CLOSE #3:IF A = ASC("P") THEN
GOTO 32700
32020 IF A = 27 THEN GOTO 40
32100 TRAP 32280
32110 DIM GRAY(3,15),BUFFERS$(400):OPEN #7,8,0,"P:"
32120 FOR Y = 0 TO 15:FOR X = 0 TO 3
32130 READ Q:GRAY(X,Y) = Q
32140 NEXT X:NEXT Y
32150 DATA 255,255,255,255,255,254,253,251,247
32160 DATA 254,222,253,219,222,219,187,183
32170 DATA 222,181,213,183,181,213,244,173
32180 DATA 181,177,213,149,177,149,212,165
32200 DATA 145,33,148,68,33,36,68,72
32210 DATA 33,1,36,8,1,4,16,64
32220 DATA 0,4,0,64,0,0,0,0
32230 BUFFERS$(1) = CHR$(27):BUFFERS$(2) = "K":BUFFERS$(3) = CHR$(
128):BUFFERS$(4) = CHR$(1)
32230 BUFFERS$(1) = CHR$(27):BUFFERS$(2) = "K":BUFFERS$(3) = CHR$(
128):BUFFERS$(4) = CHR$(1)
32240 ?#7:??#7:??#7:CHR$(27); "A":CHR$(8)
32250 FOR X = 79 TO 0 STEP -1:P = 5:FOR Y = 0 TO 191:LOCATE
X,Y,A:BUFFERS$(P) = CHR$(GRAY(INT(RND(0)*4),A))
```

```
32260 BUFFERS$(P + 1) = CHR$(GRAY(INT(RND(0)*4),A)):P = P + 2:
NEXT Y
```

```
32270 ?#7;" "":BUFFERS$:NEXT X
2280 TRAP 40000:GOTO 40
```

Anybody who has a machine language screen dump for GR.9-11 is invited to contact me.

And finally we get to the problem of using these drawings in your own programs. The most straightforward method is to use a version of the load routine from the main program (lines 1130-1190). This uses a direct call to the central input/output utility in the OS. This is explained in DE RE ATARI page 8-36.

Your program should set up a graphics 9, 10, or 11 display (with or without text window) then jump to the following subroutine (renumber as needed):

```
32000 TRAP 32040:POKE 764,12
32010 OPEN #1,4,128,FILESPEC(3RD PARAMETER SHOULD BE 0
FOR SCREENS CREATED WITH THE FIRST VERSION OF THE
PROGRAM)
```

```
32020 POKE 850,7:POKE 852,PEEK(88):POKE 853,PEEK(89):POKE
856,255:POKE 857,255
```

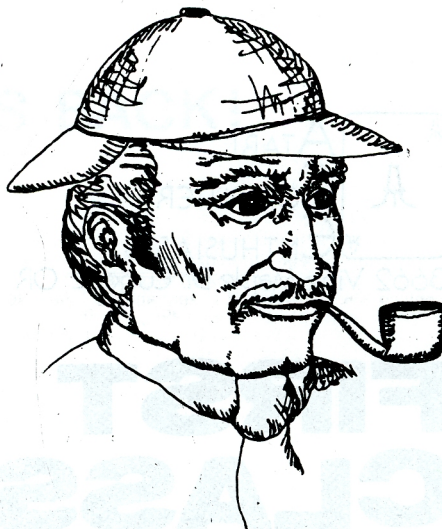
```
32030 DUM = USR(ADR("hhh*LVd"),16):CLOSE #1:REM * AND d ARE
INVERSE
```

```
32040 TRAP 40000:RETURN
```

Have your program & screen image on the same disk or in order on cassette.

I invite someone to write a routine to read a picture into data statements or a string so the image can be an integral part of the program. Perhaps a machine language routine could then move the image into screen memory. I'm interested in seeing what all of you can add to GTIA SKETCHPAD that I have missed.

[—Please send to us your pictures so we may put some of them in ACE!— Editor.]



Marc Benioff
teaches

How To Do A VBI In Assembler

Many of you are struggling with assembly language. I have recently written a routine to set up a vertical blank interrupt, and then rotate the border colors. I hope you can understand this source code, it will explain itself. Type this routine into your Assembler Editor cartridge and Assemble It. Save the program from \$600 to \$6FF. Then load it in through BASIC. Good Luck.

0 ; LOAD IN THE OBJECT CODE FROM DOS, AND THEN DO A
X=USR(1536). THIS WILL START A VBI READING FROM \$650.

```
01 ;
02 ; THIS PROGRAM CREATES A VBI
03 ; ROUTINE, AND SETS THE POINTERS.
04 ; THE POINTERS POINT AT $650
05 ; THE POINTERS ARE SET AT $600
06 ;
07 ; YOU ARE THEN ABLE TO CALL IT
08 ; FROM BASIC.
09 ;
10 * = $0650 ; PAGE 6.5
20 START LDA #0 ; PUT 0 IN A
25 LDY #0 ; PUT 0 IN Y
30 BEGIN LDX $D20A ; THIS IS THE INPUT AREA, PUT WHAT EVER IS
INPUTING HERE. $D20A IS A RANDOM # GENERATOR
40 STX $2C8 ; THIS IS THE OUTPUT AREA. PUT YOUR OUTPUT HERE.
THE EXAMPLE IS THE COLOR 4 REGISTER.
50 INY ; INCREMENT Y
60 CPY #510 ; LOOP 10 TIMES
70 BNE BEGIN ; GO TO BEGIN IF NOT 10X
80 JMP $E45F ; END OF VBI
0100 * = $0600 ; PAGE 6 x 0120 VBI = $222 ; POINTER FOR VBI
0130 LDA #START&255 ; WHERE IS THE HIGH OF START
0140 STA VBI ; STORE IT AT THE POINTER
0150 LDA #START/256 ; WHERE IS THE LOW OF START
0160 STA VBI + 1 ; STORE IT AT THE POINTER + 1
0165 PLA ; POP THE STACK
0170 RTS ; GO BACK TO BASIC, ETC.
0180 ;
0190 ; A E MEMBERS -
0200 ; I KNOW MY SOURCE COMMENTS ARE
0210 ; A BIT CONFUSING. BUT THEY
0220 ; WILL GET BETTER WITH TIME.
0230 ; THIS IS PROBABLY A LITTLE
0240 ; ELEMENTARY TO MOST OF YOU,
0250 ; BUT I HOPE IT TEACHES HOW TO
0260 ; DO A VBI FOR AT LEAST ONE.
```

SHANE ROLIN'S UTILITIES

MICROSOFT NOTE

Anyone interested in starting a MicroSort column, or anyone who has written in MicroSort for the ATARI and are willing to submit programs, tips, information, etc. to help me write one, please contact me, Shane Rolin, at 118 South Ridge Dr., Monroeville, PA 15146 or call 412-372-0616.

R-A-M-M-E-R

AHHH! So you just bought a new RAM board and you want to test it, or you've had your RAM boards, and you just want to test them out to see if they are working properly. Here is a program to test ALL free RAM. To run it, simply type it in and RUN it. If you want to test all possible free RAM, delete the REM statements. The program is self-explanatory.

Blink-Blink

This is a neat Machine-Language subroutine which makes the cursor Blink-Blink-Blink! Read the REM statements to find out how to use the BLINKING CURSOR routine.

-Have Fun!

—Shane Rolin - P.A.C.E.

Sector-Reader

(c) CSOFTWARE 1983 by SHANE ROLIN

Sector Reader is a Basic program allowing you to inspect each sector on the disk. The program is self-explanatory. There are two versions of this program, ATARI BASIC and MACHINE-LANGUAGE. The M/L version should be available from the librarian. Ask for it.

—Shane Rolin - P.A.C.E.

```
100 REM ---SECTOR INSPECTOR
110 REM ---EXAMINES DISK SECTORS
120 REM ---(C)1983 CSOFTWARE
130 REM ---16K-DISK REQ.
140 REM ---
150 REM ---BY SHANE ROLIN
160 POKE 16,64:POKE 53774,64
170 POKE 769,1
180 POKE 770,82
190 DIM A$(128),B$(10)
200 B$="h 5d":REM STRING HAS d i
n inverse and last character is a
diamond (CTRL-
210 SETCOLOR 2,0,0
220 TRAP 510
230 FOR C=1 TO 128:A$(C,C)=" ":NE
XT C:POKE 82,2:POKE 83,39
240 ? CHR$(125);"SECTOR INSPECTOR
JC (C)1983 CSOFTWARE"
250 POKE 83,34:POKE 82,3
260 D=ADR(A$)
270 E=INT(D/256)
280 F=D-(E*256)
290 POKE 772,F
300 POKE 773,E
310 POKE 770,82
320 POSITION 3,2:? "INPUT SECTOR
NUMBER":POKE 752,0
330 POSITION 3,3:? "
340 POSITION 3,3:INPUT SECTOR
350 IF SECTOR<1 OR SECTOR>720 THE
N 320
360 SECTHI=INT(SECTOR/256)
370 SECTLO=SECTOR-(SECTHI*256)
380 POKE 778,SECTLO
390 POKE 779,SECTHI
400 G=USR(ADR(B$))
410 H=PEEK(771)
420 POKE 752,1
430 POSITION 3,5:? "DISK STATUS="
;H;" (1=OK)"
439 REM ON LINE 440 AND LINE 490,
THE CHARACTER IN QUOTES IS AN E5
C-ESC
440 POSITION 3,11:FOR G=1 TO 128:
? "" ;A$(G,G);:NEXT G
450 FOR G=1 TO 128:A$(G,G)=" ":NEXT
G
460 POSITION 8,17:? "HIT ANY KEY
WHEN READY"
470 POKE 764,255
480 IF PEEK(764)=255 THEN 480
490 POSITION 3,11:? "" ;A$:POSITIO
N 3,17:? A$:POKE 764,255
500 GOTO 320
510 POSITION 3,10:? "ERROR-";PEEK
(195);CHR$(253):FOR C=1 TO 1000:N
EXT C:TRAP 32767:TRAP 510:GOTO 23
0
```


TYPESETTING FROM YOUR COMPUTER

ATARI OWNERS: If you have a modem, text editor, and communications program to send ASCII files, you should consider the improved readability and cost savings provided by **TYPESETTING** your program documentation, manuscript, newsletter, or other lengthy text instead of just reproducing it from line printer or daisy-wheel output. Computer typesetting by telephone offers you high quality, space-saving copy that creates the professional image you want! Hundreds of type styles to choose from with 8 styles and 12 sizes "on line." And it's easy to encode your copy with the few typesetting commands you need.

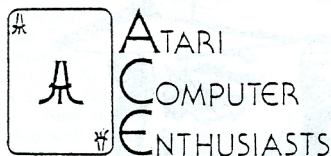
COMPLETE CONFIDENTIALITY GUARANTEED

— Bonded for your protection —

PUBLICATION DESIGN, EDITING, & PRODUCTION

Editing & Design Services
Inc.

30 East 13th Avenue Eugene, Oregon 97401
Phone 503/683-2657



3662 Vine Maple Dr. Eugene OR 97405

FIRST CLASS MAIL

Atari Computer Enthusiasts

A.C.E. is an independent computer club and user's group with no connection to the Atari Company, a division of Warner Communication Company. We are a group interested in educating our members in the use of the Atari Computer and in giving the latest News, Reviews and Rumors.

All our articles, reviews and programs come from you, our members.

Our membership is world-wide; membership fees include the A.C.E. Newsletter. Dues are \$10 a year for U.S., and \$20 a year Overseas Airmail and include about 10 issues a year of the ACE Newsletter.

Subscription Dep't: 3662 Vine Maple Dr., Eugene, OR 97405.

President—Kirt Stockwell, 4325 Sean, Eugene, OR 97402

503-689-5355

Vice Pres—Larry Gold, 1927 McLean Blvd., Eugene, Or 97405

503-686-1490

Secretary—Charles Andrews, POB 1613, Eugene, Or 974401613

503-747-9892

Librarian—Chuck and Jody Ross, 2222 Ironwood, Eugene, OR 97401

503-343-5545

Editors—Mike Dunn, 3662 Vine Maple Dr., Eugene, Or 97405

503-344-6193

—Jim Bumpas, 4405 Dillard Rd., Eugene, Or 97405

503-484-9925

E.R.A.C.E. (Educational SIG Editor)—Ali Erickson, 295 Foxtail Dr., Eugene, Or 97405

503-687-1133

Send a business-size SASE to the Ross' for the new, updated ACE Library List!!

Bulletin Board

(503) 343-4352

On line 24 hours a day, except for servicing and updating. Consists of a Tara equipped 48K Atari 400 with a TARA keyboard, 2 double-density disk drives, an Atari 825 printer, a Hayes SmartModem; running the ARMUDIC Bulletin Board software written by Frank L. Huband, 1206 N. Stafford St., Arlington, VA 22201. See the Nov '82 issue for complete details.